NASA TECHNICAL TRANSLATION NASA TT F-15,437

MULTI ACCESS CONVERSATIONAL SYSTEM (MACS)

J. Karczewski

(NASA-TT-F-15437) MULTI ACCESS CONVERSATIONAL SYSTEM (MACS) (Kanner (Leo) Associates) ~~14~~ 11 p HC $4.00 CSCL 09B

N74-19834

Unclas

G3/08 33318

Excerpt from PhD Dissertation, Date Unknown, 10 pages

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D.C. 20546 MARCH 1974

STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TT F-15,437		
4. Title and Subtitle MULTI ACCESS CONVERSATIONAL SYSTEM (MACS)		5. Report Date March 1974
		6. Performing Organization Code
7. Author(s) J. Karczewski		8. Performing Organization Report No.
		10. Work Unit No.
9. Performing Organization Name and Address Leo Kanner Associates Redwood City, California 94063		11. Contract or Grant No. NASw-2481
		13. Type of Report and Period Covered Translation
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration, Washington, D.C. 20546		14. Sponsoring Agency Code

15. Supplementary Notes

Excerpt from PhD Dissertation, Date Unknown, 10 pages

16. Abstract

The authors discusses the proposed monitor operating system for the ODRA 1204 computer, a Polish medium-capability computer which is not particularly suitable for time sharing due to its small memory. The OS takes into consideration the limitations in the hardware and the slow transfer rates and is designed to optimize the time allocated to each user in the central processor and to minimize the number of external memory transfers.

17. Key Words (Selected by Author(s))	18. Distribution Statement
	Unclassified-Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	~~12~~ 11	$4.00

NASA-HQ

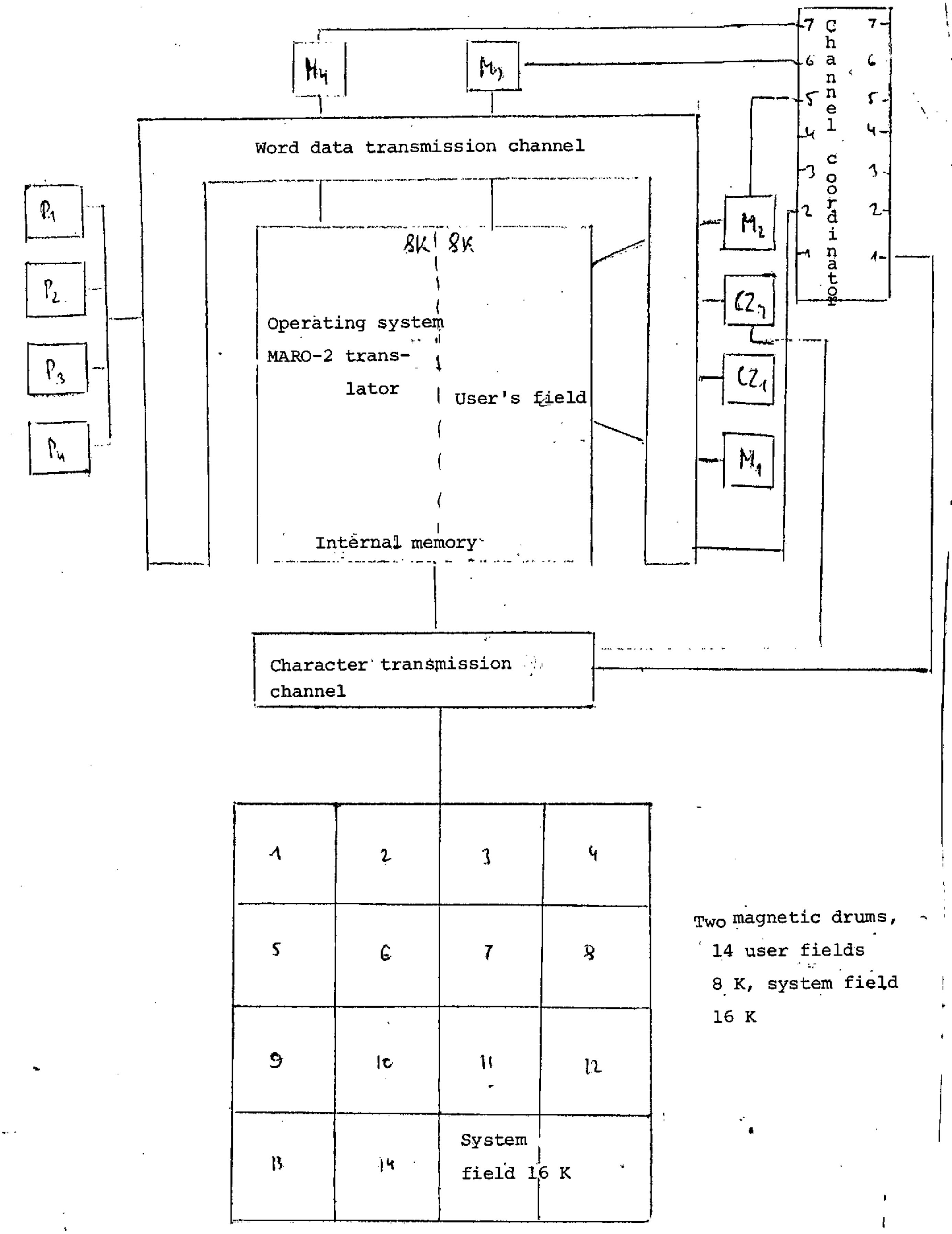
MULTI ACCESS CONVERSATION SYSTEM (MACS)

J. Karczewski

The MACS Multi Access Conversational System is an experimental multi access conversational system for the ODRA 1204 computer. This system was installed in the computing center of the Polish Academy of Sciences. The MACS enables the users to operate simultaneously from four terminal stations. Each terminal station is equipped with a monitor and perforator and two of them have, in addition, a paper tape reader. The operating system (OS) also uses a "pure" translator of the MARO-2 conversational language. Fig. 1 shows the remaining configuration of the system. /1*

The purpose of this report is to describe the proposed OS for the system that is discussed. However, before we discuss the principles on which the OS operates, we will briefly describe the main characteristics of the ODRA 1204 computer, which considerably influenced the OS concept. First, it should be emphasized that basically the ODRA 1204 computer is not intended for use in multi-access systems. The main shortcoming is the internal memory which is too small and which cannot be shared simultaneously by at least two users, assuming of course that the user's field is relatively large. The maximum field which can be assigned to the user when the memory is shared simultaneously by at least two users without segmenting the programs is 4 K. Such a field seemed to be too small to us, i.e. it would limit too much the capability of the user in taking advantage of the computer. We adopted the principle that the internal memory (IM) at a given instant should have a field for one user which is 8 K. Hence, the main problem became, because of the necessity of making the drum transfer after each /2
change in the allocation of the central processor, the minimization

* Numbers in the margin indicate pagination in the foreign text.



M_1, M_2, M_3, M_4 - Monitors ; P_1, P_2, P_3, P_4 - Perforators ; CZ_1, CZ_2 - Readers

Fig 1. System configuration.

of the time losses and related losses. By a drum transfer we shall mean the transfer of the user's field, whose allocation of the central processor (CP) has terminated to the drum memory (DM) and the transfer from the drum memory to the central processor of the user's field to whom the central processor has been allocated. The drum transfer time is considerable due to the equipment (the mean waiting time per drum revolution (the mean access time) is 20 msec, and the transfer rate is 12,000 words/sec). Therefore, the minimization of the reaction time of the system is based on:

-- the minimization of the number of drum transfers

-- the minimization of the operating time for the OS programs.

We will now present and justify the main postulates on which the implementation of the OS concept is based:

I. Instantaneous Interrupt Service

The servicing of all interrupts connected with queueing transfers and errors which do not interfere with the operation of the system is instantaneous after the interrupt has been received and protected from other interrupts. This solution was adopted in view of the short servicing time for the interrupts and also because of the resulting simplification in queueing. Using this method, the necessity of having an interrupt queue and programs which analyze such a queue has been avoided.

II. Three Operating Modes of the Operating System

The operating mode of the system depends on the use of the central processor. Therefore, we distinguish:

-- the user's operation

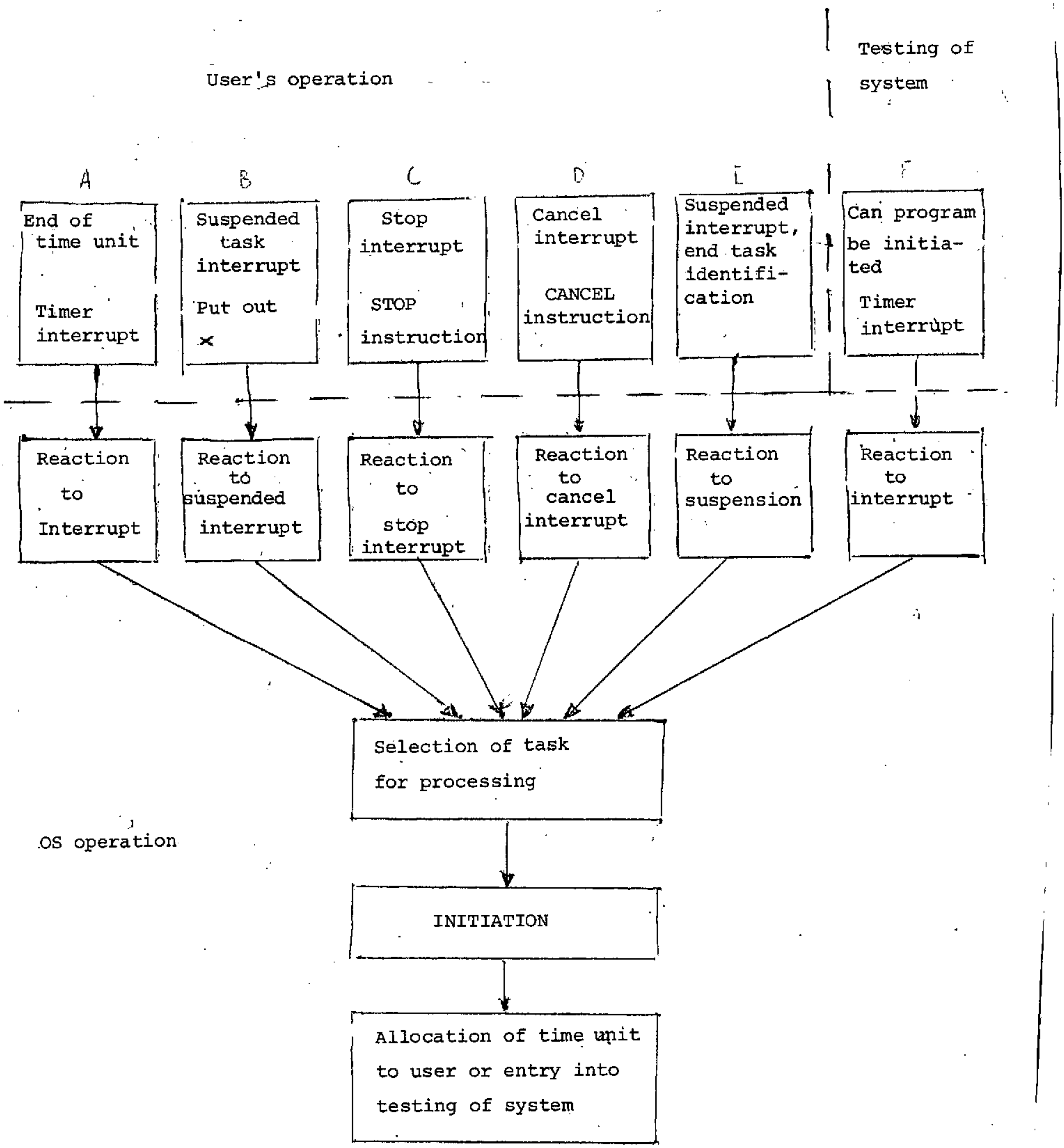
-- the OS operation

-- the testing of the system (primarily counting the "unused" time in the CP).

This makes it possible to monitor the operating mode of the system with the aid of a program monitor. Each change in the mode corresponds to a particular value of the index which indicates the operating mode. Due to this, the reaction of the system to the events which occur during the OS operation are simplified. All changes in the operating mode are strictly deterministic and occur only at particular instants determined by the OS operation (see Fig. 2). /4

III. Protection from Interrupts of a Section of the Queue Which Begins the Execution of Tasks

The events connected with tasks calling for execution (the pressing of the ZO key on the keyboard controlling the monitor by the user))are random. In order to avoid the possibility of such events occurring in time (the next task calling for execution from another or even the same terminal station before the preceding task that called for execution has been initiated) the section of the queue is executed and protected from interrupts. The execution time for this section which initiates the task is so short, that its protection from interrupts does not cause any interrupts in the OS operation. Due to this, it was not necessary to have a queue of tasks calling for execution and protect the queueing program from the next entry into it.



A, B, C, D, E, F are mutually exclusive. The termination of the OS operation causes each time a change of the operating mode from the OS operation to the user operation or the testing of the system.

Fig. 2. General flow diagram for the OS operation.

IV. Queueing

The queueing of the tasks in the MACS OS system is based on the FIFO (first in, first out) principle. Due to the small number of tasks that can be executed simultaneously (four), the queue has a linear structure. The sequence in the CP as follows:

first location	number of tasks in queue	/5
second location	first task (number of task)	
*	second task (number of task)	
*	third task (number of task)	
fifth location	fourth task (number of task)	

If there are fewer than four tasks in the queue, the remaining locations are zeroed out. The operations carried out on a queue set up in this manner (the adding of a task to the queue, the removal of a task from the queue, cyclic permutations of the task (the first tasks becomes the last and checks whether a given task is in the queue) are simple and do not require a long time (several machine operations).

Before we begin the discussion of the OS block diagrams, we will focus attention on two matters: the principle for allocating the CP to particular tasks and the interrupt system. As we already mentioned, one of the most important problems is the minimization of the number of drum transfers, clearly with the condition that it have no negative effect on the reaction time of the system. We adopted the following principle: we extend to the user the allocation of the central processor as long as it does not have a detrimental effect on the reaction time of the system. This is based on the fact that the OS always behaves as if there were four users in line. Hence, each user must wait three time units. Due to this, for example, for two users, the allocation of the central processor to each of them consists of three time units, not one unit, which makes it possible to considerably reduce the number of drum transfers. Another important aspect is the system of interrupts on which the functioning of the multi-access system is based. We took advantage both of hardware and program interrupts. We will describe the most important simultaneous roles which they fulfill in the functioning of the OS.

1) Hardware Interrupts: /6

-- timer interrupt

The frequency of the timer interrupts depends on the operating mode of the system. In the OS operating mode, we want the interrupts to occur as rarely as possible since they prolong the OS operating time (interrupts up to 640 msec).

In the user's mode the interrupts determine the length of the time unit allocated to the central processor. Two interrupts occur in one time unit, i.e. it can be from 4 msec to 1.28 sec. The length of the time unit can be changed each time during the initiation of the operation of the system (beginning of a new operational cycle). The change in the length of the time unit will be completed if the collected statistical data indicate that the efficiency of the system is improved as a result of this. In the operating mode used for the testing of the system, we want the most frequent interrupts, since they enable us to react most rapidly to the events occurring in the system.

(Interrupts every 2 msec).

-- monitor interrupts,

which provide information about the operation of monitors are primarily important for interrupts connected with monitor transfers and also provide information about errors in the monitors,

-- interrupts of internal equipment: readers and perforators, provide information about the operation of the interal equipment, the transfers and errors in the equipment;

-- interrupts connected with the operation of the magnetic drum provide information about the drum transfers and errors in the drum.

2) Program Interrupts (Software Interrupts)

-- suspended interrupt

The suspended interrup occurs as a result of the execution of the current task. The task is interrupted from the instant /7 when the OS puts out an "x" on the monitor. This is the state when it expects the next task from the user.

-- stop interrupt

The task is stopped when the OS executes the STOP instruction put in by the user.

-- cancel interrupt

The task is cancelled when the OS executes the CANCEL instruction put in by the user of the system. Fig. 2 shows the block diagram for the operation of the OS. It is worthwhile to emphasize that entry into the OS programs is possible under mutually exclusive situations. The testing excludes the user's operation. The, suspended interrup stop interrupt and cancel interrupt are mutually exclusive and also eliminate the possibility of signaling the end of the time unit. The entry into the operating programs after identification was completed is related to the necessity to prepare the field for the user who begins the operation in MARO-2. The identification does not require an assignment of the field to the user.

Fig. 3 shows the detailed OS operational block diagram. The block diagram shows in which way the postulate for minimizing the number of drum transfers was taken into account and on what the determinism in the OS is based. The main characteristic feature is the initiation of the programs in the OS at a particular designated instant. Due to this the instant when a task is added to the queue for the central processor is delayed relative to the instant when a task is ready to be initiated. However, from the standpoint of the user, this delay does not matter, since it is added before the memory in the central processor is allocated to it. At the end, we give the initiation block diagram (Fig. 4).

The adopted system configuration as well as the specific characteristics of the ODRA 1204 computer considerably influenced the planned operating system. It was based on certain basic requirements which are related to this. When the proposed system was discussed, a great deal of attention was given to practical ways of minimizing the effect of the limitations of the equipment. A rich statistical collection was incorporated in the system which we hope will prove the usefulness of the configuration that was adopted and the proposed OS from the standpoint of the effectiveness of the system.